DELIVERABLE D2.1

TASK-BASED LINGUISTIC INTERACTION

Amy Isard (UEDIN)

Beneficiaries: UEDIN (lead), HWU, FORTISS, UNIBI
Workpackage: WP2: Natural Language Communication
Description: The general objective of WP2 is to develop components for recognising, understanding, and generating embodied natural language. In this report we describe the natural language processing components included in the initial integrated system, which can recognize and generate simple task-based utterances.

Version: Final
Nature: Report (R)
Dissemination level: Public (PU)
Pages: 8
Date: 2012-02-29
1 Introduction

In JAMES, natural language plays an important role in both the input processing and in the selection of behaviours for the robot agent, since natural-language interaction is central to social communication. The general objective of WP2 is to develop components for recognising, understanding, and generating embodied natural language. This report describes the components which we have developed for the first version of the integrated JAMES system (see D7.1) which has been used for pilot user testing and evaluation.

The report covers the natural language components in turn. First we describe the components which deal with recognising the user's spoken input: the speech recognition system (section 2), and the natural language interpretation (section 3). Then we turn to the components which create the system's spoken output and pass it on to the actuators along with the outputs for the other modalities: the natural language generation (4) and the multimodal output generator (5).

2 Automatic Speech Recognition

We have chosen to use the Microsoft Kinect and the associated Microsoft Speech API for the speech recognition component of JAMES. The hardware consists of a high-quality microphone array, and the software provides a list of hypotheses for each speech utterance. At the time of the first system evaluation, the Microsoft Speech API for Kinect was only available in English, but Microsoft state that other languages will be released in the near future. Another feature of the Kinect is that the microphone array allows us to gather information about the direction from which the speech is coming, which in future multi-person scenarios can be combined with the vision data to decide which customer is speaking. We created a speech recognition grammar in the SRGS format which covers all of the expected customer utterances for the first system evaluation. In this first version of the system, it was assumed that only one person would be speaking at a time.

2.1 Speech Recognition Grammar

We created a grammar using the WC3 standard Speech Recognition Grammar Specification Language (SRGS) [Hunt and McGlashan, 2004]. This allows us to constrain the recognition task, and achieve more reliable recognition results. Figure 1 shows a fragment of the grammar, which allows the recognition of a number of sentences including:

Can I get a coke please
Could I have a juice
A juice please

Each rule element specifies items which can contain one or more words, which are then combined in the top level item element. If a rule has the attribute repeat="0-1", this means that it is an optional part.

2.2 Speech Recognition Output

The Kinect provides hypotheses while it is performing speech recognition, and then a final best hypothesis, with an integer expressing the recognition confidence. We currently have a fixed thresholds below which we consider the speech recognition to be unreliable, and we plan to experiment with this threshold in experimental condition in future pilot experiments to find the optimum value. We are currently using the best hypothesis, and passing this on to the Natural Language Interpretation module along with the confidence, the angle of the sound source, and a confidence value for the angle. We have defined an ICE object which contains all of the speech recognition output data, which is passed to the Natural Language Interpretation Module.
Figure 1: SRGS grammar
3 Natural Language Interpretation

Once the user speech has been recognised, it must be further processed to extract the underlying meaning. The natural language interpretation module receives the speech recognition hypothesis, and parses it to provide syntactic and semantic information, which is then passed on to the multimodal fusion component. As well as extracting the meaning of the user’s utterance, we also recognize their politeness level (for instance, whether they have used the word “please”) - this is then passed through the system and can later influence the politeness of the output language.

3.1 OpenCCG Grammar

We built a bi-directional OpenCCG grammar [White, 2006] which is used for both parsing and generation. The grammar contains both syntactic and semantic information. Words defined in the dictionary are assigned both a part of speech and a “family” in the lexicon, and optionally also to a class in a type hierarchy which can be used to constrain parses to particular classes. A fragment of the type hierarchy is show in figure 2. It is possible to assign multiple words to one stem, which allows variability in system output, but also provides useful information to pass on to the fusion module. For instance in figure 3 “hi” and “hello” have both been assigned to the stem “greet”. An OpenCCG family provides the grammatical rules, and adds semantics and an example is shown in figure 4. This specifies that please can occur before an interrogative sentence which does not already have the feature “politeness”, and it then adds the politeness feature and creates a semantic representation.
Figure 4: Fragment of OpenCCG family for “please”
4 Natural Language Generation

JAMES's linguistic behaviour is based on carrying out high-level communicative acts selected by the high-level planner (WP4). The output focuses on domain actions such as requesting and clarifying the users' needs, as well as social actions such as managing attention.

In JAMES we are using OpenCCG to generate the robot language output, and the OpenCCG grammar described in section 3 also contains all of the language necessary for the robot's speech output. In collaboration with WP4 we created an API which represents the language output in terms of communicative acts, based on Rhetorical Structure Theory (RST). The natural language generation module then translates the RST structures into OpenCCG logical forms, which are sent to the OpenCCG realizer which outputs text strings. The RST is described in detail below.

4.1 RST for language output

The output to the generation components of spoken dialogue systems have typically specified output in terms of simple dialogue acts, such as assert (predicate) and direct (predicate), where the predicate is a basic representation of semantics as a predicate-argument structure. In order to have context fully taken into account in the output language, therefore, the language generation system needs to have full access to the dialogue context, and may need to take dialogue-specific issues into account in planning an utterance. In JAMES we are using a much more flexible interface between the planner and NLG, effectively moving the first text planning stage inside the planner. Thus the planner does the content selection and then produces XML representations of RST structures.

The RST language for JAMES is based on similar structures used in the FP6 JAST project (IST-FP6-003747-IP). Dialogue acts are divided into four types, each of which has one or more subtypes, as shown in figure 6. In the XML representation which we use for the API, each dialogue act is represented by a speech element, and its subtype by a pred element. We also include a person element to specify which customer is being addressed, and object elements if appropriate to refer to drinks. Some examples are shown in figure 7.

We plan to submit a paper describing our use of RST to describe dialogue acts for natural language generation to SemDial 2012, the 16th Workshop on the Semantics and Pragmatics of Dialogue (https://sites.google.com/site/semdial2012seinedial).
• query
  – offer

• inform
  – list
  – not-understand
  – hand-over

• reply
  – positive-reply
  – negative-reply

• social
  – greet
  – bye
  – thanks
  – acknowledge-thanks
  – acknowledge

Figure 6: RST dialogue act types

"hello"
<speech type="social">
  <person id="a4"/>
  <pred type="greet"/>
</speech>

"what would you like to drink?"
<speech type="query">
  <person id="r35"/>
  <pred type="offer">
    <object type="drink"/>
  </pred>
</speech>

"here is your juice"
<speech type="inform" politeness="4">
  <person id="8fhe47"/>
  <pred type="hand-over">
    <object type="drink" name="juice" id="A1"/>
  </pred>
</speech>

Figure 7: Examples of RST for dialogue acts
5 Multimodal Presentation Planning

In addition to speech, the robot system will also express itself through facial expressions, gaze behaviour, and actions of the robot manipulators. The presentation planner coordinates the output across the various multimodal channels to ensure that it is coordinated both temporally and spatially, using techniques similar to those employed in previous EU projects such as JAST (FP6-003747-IP) and COMIC (IST-2001-32311).

We have created an XML format which contains specifications for all of the output modalities including the RST described in section 4.1. The robot talking head can currently express a number of pre-assigned expressions, and the robot arm can hand over a drink to a customer (see WP6). The multimodal planner generates text from the RST as described in section 4 and then sends each output to the appropriate actuator.

The example in figure 8 would cause the robot to smile while handing a juice to the customer and saying “here is your drink”.

References
